Appl. Math. Lett. Vol. 4, No. 1, pp. 13–15, 1991
Printed in Great Britain.

0893-9659/91 $3.00 + 0.00

A Full Parallel Preconditioner for a Class of M-Matrices

L. BRUGNANO AND D. TRIGIANTE

Dipartimento di Matematica, Universita' di Bari

(Received October 1989)

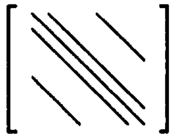
Abstract. A full parallel preconditioner for preconditioned conjugate gradient (PCG) methods is derived, by using an incomplete cyclic reduction, for a class of M-matrices.

1. INTRODUCTION

Let us consider the linear system:

$$Ax = b. \tag{1.1}$$

to be solved with a PCG method. Let A be a five-diagonal, symmetric, weakly diagonally dominant M-matrix, with structure:

A =  **NxN** . (1.2)

The extreme diagonals have distance k from the main one ($N = k^2$). Let us suppose $k = 2^r$.

If P_N is the permutation matrix which takes first the odd rows and then the even ones, one has:

$$P_N \cdot A \cdot P_N^T = \begin{bmatrix} T_1 & S^T \\ S & T_2 \end{bmatrix}, \tag{1.3}$$

with $T_1, T_2, S \in \mathbf{R}^{N/2 \times N/2}$. Moreover, the blocks T_i are symmetric tridiagonal, with band-size $k/2 = 2^{r-1}$, diagonally dominant, while S is upper bidiagonal. Let us consider the block LU factorization of (1.3):

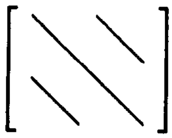
$$P_N \cdot A \cdot P_N^T = \begin{bmatrix} T_1 & 0 \\ S & B_1 \end{bmatrix} \cdot \begin{bmatrix} I & T_1^{-1} \cdot S^T \\ 0 & I \end{bmatrix},$$

with

$$B_1 = T_2 - S \cdot T_1^{-1} \cdot S^T. \tag{1.4}$$

The matrix B_1 is no more a sparse one. To obtain a parallel preconditioner for a PCG method, one can use a sparse approximation of B_1. For example, we can choose a five-diagonal one, $\tilde{B}_1$, with the same structure of A (see (1.2)), but with dimension and band-size halved, by suppressing the other diagonals. It is known that B_1 and $\tilde{B}_1$ are strongly diagonally dominant M-matrices (see [2]).

If this is the case, we can reduce $\tilde{B}_1$ by repeating the above operations. After j steps, we choose $\tilde{B}_j$ as the tridiagonal matrix:

$\tilde{\mathbf{B}}_j$ **=** 

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX of band-size 2^{r-j} (this is justified from the fact that the terms on the other diagonals are smaller). To solve the linear system in which $\tilde{B}_j$ is the coefficient matrix (this is necessary to get the preconditioned residual), one transforms $\tilde{B}_j$ to $P_{N_j}^{r-j}\tilde{B}_j(P_{N_j}^{r-j})^T (N = 2^{2(r-j)})$. The resulting matrix is a block diagonal one, with 2^{r-j} blocks, each one tridiagonal with band-size 1.

From this fact, it is self-evident that one can solve the linear system to get the preconditioned residual on 2^{r-j} processors with constant degree of parallelism, or on 2^{r-i} processors, for some $i = 1, 2 \dots, j$, with degree of parallelism increasing from 2^{r-j} to 2^{r-i}. The number of processors used can be doubled, using a twisted factorization of the tridiagonal blocks (see [1] or [5]), with a little added overhead. Such a PCG algorithm is, therefore, fully parallel.

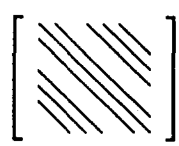
2. Evaluation of $\tilde{B}_1$

To compute $\tilde{B}_1$, we need only the main three nonzero diagonals of T_1^{-1}. Generalizing the results in [3], one can derive an algorithm to obtain T_1^{-1} by diagonals. Moreover, one can show that the elements on the diagonals becomes smaller and smaller, as we go away from the main one. If we call $\tilde{T}_1^{-1}$ the considered part of T_1^{-1}, we get (see (1.4)):

$$\tilde{B}_1 \cong T_2 - S \cdot \tilde{T}_1^{-1} \cdot S^T. \qquad (2.1)$$

Actually, one can show that

$$T_2 - S \cdot \tilde{T}_1^{-1} \cdot S^T =$$



, (2.2)

while the structure of $\tilde{B}_1$ is the one in (1.2). It follows that some of the information of T_1^{-1} is neglected: in fact the derived preconditioner (BT0) is not satisfactory. Slightly better results are obtained summing the neglected diagonals of (2.2) on the main one (BT1). The modified version (MBT1), obtained imposing that B_1 and $\tilde{B}_1$ must have the same row sum, works very bad. The best performances are obtained by imposing that T_1^{-1} and $\tilde{T}_1^{-1}$ (see (1.4) and (2.1)) must have the same row sum: practically, the neglected diagonals of T_1^{-1} are summed to the main one of $\tilde{T}_1^{-1}$. The resulting algorithm is denoted by MBT0.

3. Numerical Results

The numerical tests here reported are obtained from the discretization of the problem: $-\Delta u = f$, on $\Omega = [0,1]x[0,1], u|_{\partial\Omega} = 0$.

The preconditioners are tested by using as step of discretization $h = (2^r + 1)^{-1}$, $r = 3, 4, 5, 6, 7$. The resulting linear systems have dimension $N = 2^{2r}$. The starting point is $x_0 = 0$, while the stopping criterion is $||r_i||_2/||r_0||_2 < 10^{-6}$, being r_0 the initial residual, and r_i the one to the ith step.

The tests were carried out on a single processor, for the following algorithms: conjugate gradients (CG), PCG with incomplete Cholesky (IC) (Eisenstat's implementation [4]), PCG with modified incomplete Cholesky (MIC), PCG with incomplete block Cholesky (INV), PCG with modified incomplete block Cholesky (MINV), BT0, BT1, MBT0, MBT1, with three steps of reduction.

In the following table the computatinal cost, in terms of number of operations per iterate and required memory, is reported.

Method	Operations	Memory
CG	20N	7N
IC / MIC	24N	11N
INV / MINV	36N	10N
BT0 / BT1 / MBT0 / MBT1	41N	12N

In the last table the number of iterations and the work/N is reported, for the values of r considered.

Method	r = 3	r = 4	r = 5	r = 6	r = 7
CG	10 - 200	22 - 440	44 - 880	90 - 1800	179 - 3580
IC	8 - 192	13 - 312	23 - 552	42 - 1008	71 - 1704
MIC	8 - 192	12 - 288	17 - 408	24 - 576	36 - 864
INV	5 - 180	7 - 252	12 - 432	20 - 720	36 - 1296
MINV	5 - 180	7 - 252	11 - 396	16 - 576	22 - 792
BT0	6 - 246	9 - 369	16 - 656	29 - 1189	55 - 2255
BT1	6 - 246	8 - 328	14 - 574	23 - 943	44 - 1804
MBT0	7 - 287	10 - 410	14 - 574	18 - 738	28 - 1148
MBT1	7 - 287	11 - 451	19 - 779	37 - 1517	96 - 3936

From these results it seems that some of the preconditioners here introduced (in particular MBT0) can be very effective on a parallel computer. For example for $r = 7$ one can use 32 processors obtaining a theoretical speed-up (MBT0 with respect to MINV) given by $S_{32} \cong 22$.

References

1. L. Brugnano, Two-level twisted preconditioning for parallel computers, submitted.
2. P. Concus, G.H. Golub and G. Meurant, Block preconditioning for the conjugate gradient method, *SIAM J. Sci. Stat. Comp.* **6**, 220–252 (1985).
3. P. Concus and G. Meurant, On computing the INV block preconditioning for the conjugate gradient method, *BIT* **26**, 493–504 (1986).
4. S. Eisenstat, Efficient implementation of a class of preconditioned conjugate gradient methods, *SIAM J. Sci. Stat. Comp.* **2**, 1–4 (1981).
5. H.A. Van der Vorst, Analysis of a parallel solution method for tridiagonal linear systems, *Paral. Comp.* **5**, 303–311 (1987).

Dipartimento di Matematica, Universita' di Bari, Italy